A semantic SLAM system for dynamic environments

Fangchao Hu*, Qian Zong, Xurong Mou, Yin Chen, Han Wang, Miao He

College of Mechanical Engineering, Chongqing University of Technology, Chongqing, China

ABSTRACT

With the development of robotics and self-driving vehicles, simultaneous localization and mapping technology has become critical. Visual SLAM utilizes the camera as the primary sensor and shows significant advantages in texture-rich static environments. However, conventional vision SLAM systems do not perform well in dynamic environments, such as changes in lighting, loss of camera positioning, or movement of feature points due to object motion. This study proposes a new SLAM system for dynamic environments, YOEC-DSLAM. YOEC-DSLAM combines a lightweight semantic segmentation network and a multi-view geometry approach to effectively identify and filter dynamic feature points, thereby significantly improving the accuracy and robustness of map construction. These improvements improve the real-time and accuracy of the system across the board. Experiments show that YOEC-DSLAM exhibits unique advantages in processing highly dynamic scenes.

Keywords: SLAM, Lightweight networks, Semantic segmentation, Dynamic scenes

1. INTRODUCTION

Simultaneous Localization and Mapping is a key technology that helps robots operating in homes [1], [2], [3] self-driving vehicles [4] in factories, and blind people [5] to locate and build maps of their environments. This technology helps entities to locate, understand and navigate in their environments. In order to perform these tasks, a computer vision problem, visual SLAM, must be solved. Previously, the input data for building vision SLAM were sonar sensors, 2D laser scanners, and LIDARs. When the sensor used are LiDARs, the results are more accurate, but LIDARs are more expensive than image sensors[6]. With the advancement of time, the development of computer hardware and image sensors has brought many new and cheaper types of data, such as monocular, stereo and RGB-D. They are allowed for the collection of more visual information about the surrounding environment making the construction of maps more accurate.

Environments can be categorized as static and dynamic [7]. Moving objects in dynamic environments can significantly degrade the localization and mapping capabilities of SLAM systems.

Early scholars usually used traditional computer vision methods for dynamic point detection to reduce the interference of moving targets. Such methods typically detect moving object regions by analysing and modelling the motion of pixels or feature points with image sequences. Kundu et al.[8] defined geometric constraints by constructing a basis matrix that is considered dynamic if the matched feature points in subsequent frames are far from the poles. Zou and Tan [9] calculated the re-projection error of feature tracking by analysing the triangular sectional coherence and projecting the feature points from the previous frame to the current frame, projection error. If the error is large, the map points are considered to be dynamic. Wang et al. [10] designed a statistical feature-based moving target recognition model by clustering the depth image into several objects, calculating the number and percentage of feature points on each object, and eliminating all feature points on the model based on whether the target is considered to be moving. However, such methods cannot effectively acquire semantic a priori information of individual pixels, which may lead to detection errors or omissions, especially in environments with complex motion patterns, noise, occlusion, and dynamic lighting conditions. In a summary, such methods cannot provide accurate dynamic object detection.

With the advancement of deep learning technology, some scholars have started integrating deep learning methods into SLAM systems. For instance, methods like DynaSLAM[11], DS-SLAM[12], and Detect-SLAM[13] utilize target detection and semantic segmentation networks like MASK-RCNN[14], SegNet[15], and SSD[16], respectively.

*corresponding author: fangchaohu@cqut.edu.cn

International Conference on Automation and Intelligent Technology (ICAIT 2024), edited by R. Usubamatov, S. Feng, X. Mei,
Proc. of SPIE Vol. 13401, 134010N ·
doi: 10.1117/12.3053120

These methods filter out dynamic regions through semantic segmentation, thereby improving the robustness of SLAM systems. However, these methods typically require substantial computational resources, becoming a challenge for resource-constrained mobile robots.

To overcome these challenges, this study introduces a lightweight design of the YOLOv5-SEG semantic segmentation algorithm to meet real-time requirements. We design a lightweight backbone network and add attention mechanism modules in the network's neck part to enhance segmentation accuracy. Additionally, based on the ORB-SLAM2[17] framework, We combine a lightweight semantic segmentation network and a geometric module to filter out moving objects in a dynamic scene, thus building a semantic SLAM system for dynamic environments. Ultimately, we achieve the effect of accuracy and real-time improvement.

The main contributions of this paper are as follows:

Propose a lightweight semantic segmentation network, YOEC-SEG. We designed a lightweight backbone network based on YOLOv5s-SEG and introduced GhostConv, SimSPPF, and CBAMC3 modules to optimize the network structure, achieving high accuracy and real-time performance.

Combine semantic segmentation and SLAM for dynamic environments. We integrated the lightweight semantic segmentation network into the ORB-SLAM2 framework and proposed a dynamic feature point filtering strategy based on multi-view geometry, effectively improving the accuracy and robustness of map construction in dynamic environments.

Evaluate the system on public datasets. We conducted experiments on the TUM RGB-D dataset to demonstrate the effectiveness of YOEC-DSLAM in dynamic environments. The results showed significant improvements in trajectory accuracy and real-time performance compared to existing SLAM systems.

2. YOEC-DSLAM

In this section, YOEC-DSLAM is described in detail. This section contains three aspects. First, the architecture diagram of the YOEC-DSLAM framework is presented. Second, the lightweight real-time semantic segmentation method used in YOEC-DSLAM is briefly introduced. Finally, we propose a filtering algorithm for dynamic feature points in dynamic scenes.

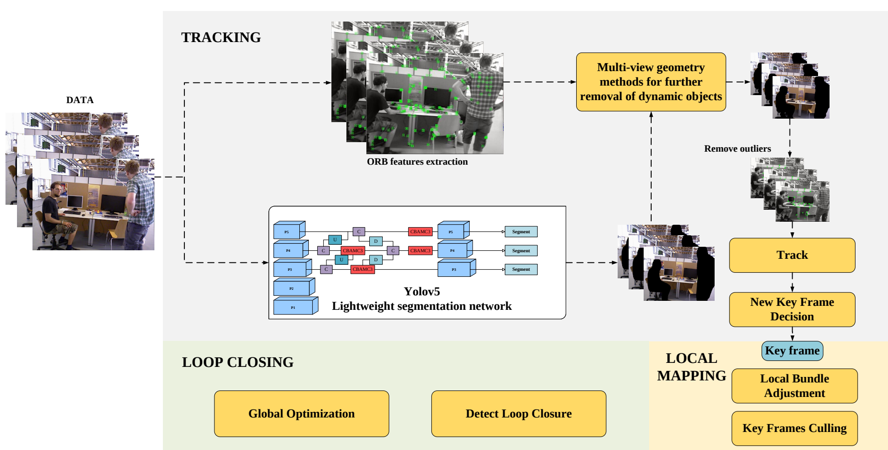


Figure. 1. YOEC-DSLAM.

2.1 YOEC-DSLAM system

In practical applications, accurate position estimation and reliability in harsh environments are key factors in evaluating autonomous robots. ORB-SLAM2 performs well in most practical situations. Therefore, YOEC-DSLAM uses ORB-SLAM2 to provide a global feature-based SLAM scheme. The image is first processed using a lightweight semantic segmentation network to obtain the semantic information of the image. Then a mask image of dynamic objects in the image is generated based on the semantic information to remove the feature points present on the dynamic objects in the image. Feature points on the edges of the dynamic object and feature points present but not detected on the dynamic object are then further detected and removed using a multi-view geometry approach. Eventually the feature points used for tracking are all located on the static objects. The system framework is shown in Fig.1.

2.2 YOEC-SEG segmentation of dynamic objects

To achieve real-time and accurate pixel-level dynamic object segmentation, we have made three key improvements to the YOLOv5s-SEG backbone network, aiming to enhance the algorithm's performance and also lightweighting the network structure. The system framework is shown in Fig.2.

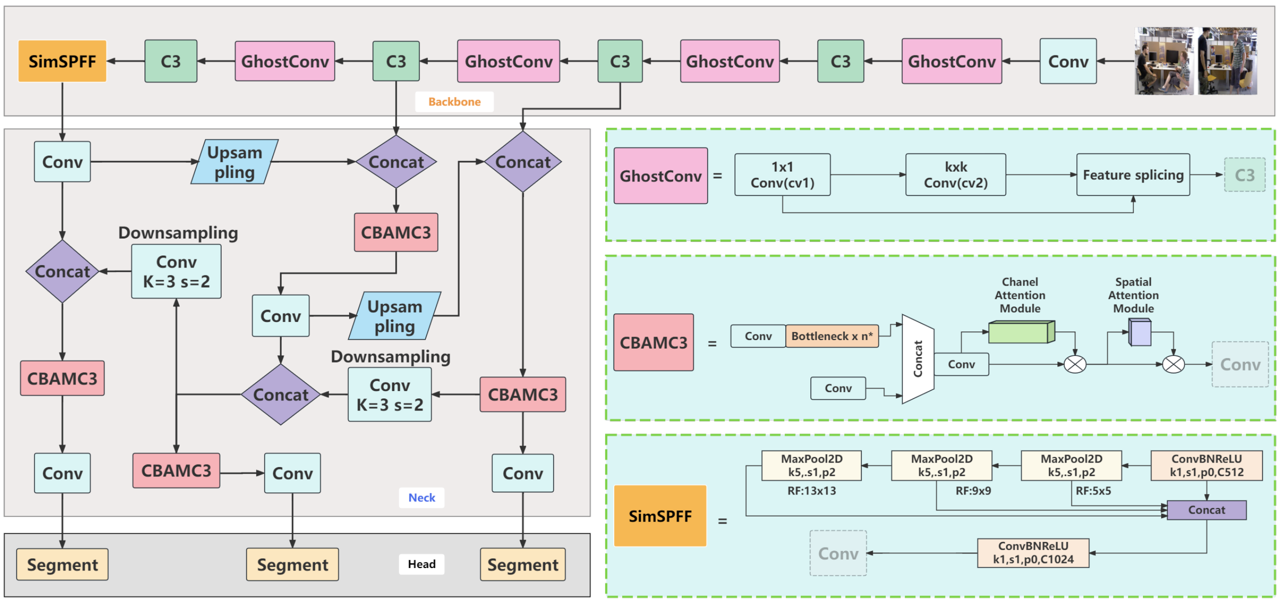


Figure. 2. YOEC-SEG Network.

First, we optimize the backbone of the CSPDarkNet network, an efficient convolutional neural network that reduces computational complexity while maintaining feature richness through the CSP (Cross Stage Partial) technique. In the improvement, all the Conv modules are replaced with GhostConv modules except the first Conv module in the original network is retained to maintain the integrity of the feature extraction. GhostConv is an efficient convolutional operation that reduces the number of parameters by generating a virtual feature map, thus significantly reducing the computational cost.

Given an input $F \in R^{c\times w\times h}$ (where c is the number of channels, h is the height, and w is the width), the convolution kernel that passes through $n \times k \times k$ to get the feature map $\mathbf{F'} \in \mathbf{R^{n\times h'\times w'}}$. The number of parameters for standard convolution P_1 is shown in Equation (1).

$$P_1 = n \times c \times k \times k \tag{1}$$

Computational complexity of standard convolutions C_1 is shown in Eqs.(2).

$$C_1 = h' \times w' \times n \times c \times k \times k \tag{2}$$

The parameter count of GhostConv P_2 is shown in Eqs.(3).

$$P_2 = \frac{n}{s} \cdot c \cdot k \cdot k + (s-1) \cdot \frac{n}{s} \cdot d \cdot d \tag{3}$$

Computational complexity of GhostConv C_2 is shown in Eqs.(4).

$$C_2 = \frac{n}{s} \cdot h^{'} \cdot w^{'} \cdot c \cdot k \cdot k + (s-1) \cdot \frac{n}{s} \cdot h^{'} \cdot w^{'} \cdot d \cdot d \tag{4}$$

It is evident that both the parameter count and computational complexity of GhostConv are significantly lower than those of standard convolutions.

In addition, we replace the SPPF module in the original network with the SimSPPF module, which employs the ReLU activation function and retains the parallel structure of the SPP to improve computational efficiency and retain more feature information.

Secondly, we optimize the network's feature extraction structure. In the Neck section, the original C3 (CSP bottleneck with 3 convolutions) module is replaced by the CBAMC3 module, a C3 module combined with a CBAM (Convolutional Block Attention Module) module. The C3 module plays a significant role in increasing the network depth and receptive field in the YOLOv5 network. The CBAMC3 module, on the other hand, introduces the CBAM attention mechanism, which further enhances the model's ability to process channel and spatial information. CBAM integrates channel and spatial attention mechanisms, enabling the model to focus on important feature information by learning channel weights and creating spatial attention maps. The addition of CBAM enables the network structure to extract richer and more comprehensive feature representations. Fig.3 visualizes the effect of the semantic segmentation network.

Figure. 3. Effects of YOEC-SEG network (a) Original image data without YOEC-SEG semantic segmentation algorithm (b) Image data processed with YOEC-SEG semantic segmentation algorithm.

2.3 Dynamic feature point filtering strategy

In order to achieve an efficient global feature SLAM system, ORB feature points are extracted from the RGB image input by the ORB-SLAM2 system. Subsequently, the image is processed using a lightweight semantic segmentation network to generate category labels for each pixel. A set of masks is generated based on these labels to represent dynamic objects. Then, the feature points on the dynamic objects in the image are removed based on the mask set to get the filtered dynamic feature points.

With the improved network, most dynamic objects can be effectively segmented and excluded from tracking and map building. However, for objects that have no priori information and moving, our semantic segmentation network may not be able to complete dynamic feature point removal. To address this challenge, we employ a multi-view geometry approach to further segment these dynamic objects.

Once the underlying dynamic content is segmented, we utilise the static feature points in the image to track the camera pose. Specifically, we will project map feature points in image frames and find correspondences in static regions of the image to optimise the camera pose by minimising the reprojection error. For each input image frame, we select multiple keyframes with the highest overlap with the current frame for computation based on the results of semantic segmentation to balance accuracy and computation[18]. When the number of frames is insufficient, we rely only on the results of semantic segmentation to remove dynamic feature points. According to the result of semantic segmentation, the points outside the semantic segmentation mask of dynamic objects are considered as static points and the points inside the mask are considered as dynamic points. The static points are first used for preliminary tracking to obtain a more accurate position. For each feature point x in the selected keyframe, based on the map obtained from the preliminary tracking, its projection point $x^{'}$ in the current frame is calculated as well as the projection depth d_{cf}.

$$d_{cf}x' = d_{kf}Rx + t \tag{5}$$

The summation in Eqs.(5) d_{cf} and d_{kf} represent the depth of projection of a point X in space on the two images, and $x^{'}$ and x are the normalised coordinates of X on the two images.

Figure. 4. Dynamic feature point screening strategy. (a) All extracted feature points (b) Static feature points remaining after removal by the dynamic feature point algorithm.

After the above processing, the remaining feature points are regarded as static points and they will be used for new position estimation and tracking tasks. The dynamic feature point removal results are shown in Fig.4.

3. EXPERIMENTAL RESULTS

3.1 Experimental environment

All experimental parts of this paper are conducted on a Desktop PC with an Intel Core i5-10400 @2.90GHz processor, 16GB of RAM, and an NVIDIA GeForce RTX 2080Ti. The operating system is Ubuntu 18.04, and the deep learning framework used is Pytorch 1.10.1. Python 3.8 is used as the programming language, and ROS Melodic is installed. The parameters of experimental equipment is shown in the Tab. 1.

Table 1: Experimental equipment

Equipment	Name
hardware enviroment	CPU: Intel Core i5-10400 @2.90GHz; RAM: 16GB DDR4 3200mhz; GPU: NVIDIA GeForce RTX 2080ti
software enviroment	Ubuntu18.04; Pycharm2023
network framework	CUDA11.4; PyTorch1.10.1
programming language	Python3.8

3.2 Datasets

In this study, MS COCO[19] dataset is used to train and evaluate the proposed YOEC-SEG network, a deep learning model for semantic segmentation. The COCO dataset is a rich dataset widely used in the field of object detection, segmentation, captioning, and contains more than 330,000 images, 200,000 of which are with detailed annotations, covering 80 categories and more than 1.5 million individuals. These images are mainly derived from complex scenes, and the locations of the objects are labelled by accurate segmentation techniques.

We use the TUM RGB-D [20] dataset to evaluate the accuracy and robustness of our improved SLAM system. The TUM RGB-D dataset contains 39 sequences recorded by Microsoft Kinect sensors in different indoor scenarios covering a wide range of SLAM-related tasks. Some of these sequences contain highly dynamic objects, such as walking figures, which pose challenges to SLAM systems. We evaluated the YOEC-DSLAM system using the TUM RGB-D dataset and compared it with ORB-SLAM2 to quantify our performance improvement in dynamic environments.

3.3 YOEC-DSLAM experimental results

By comparing the results of the first three sets of experiments in Tab. 2, we find that the model achieves the highest prediction accuracy of 86.9%, along with the smallest GFLOPs of 24.6, only when the C3 module in the Neck part is replaced by the CBAMC3 module.However, the number of parameters is reduced by only 9.5% and the model size is reduced by only 9.6% when the highest prediction accuracy is achieved. To further reduce the model size, we introduced the SimSPPF module. The introduction of this module reduced the number of parameters by 18.1%, the model size by 17.9%, the GFLOPs by 2.3, and the inference time by 64.1%.

It can be seen that these improvements resulted in a significant increase in the lightness of the model. Compared to the original YOLOv5s-SEG model, the improved model increased the prediction accuracy by 28.2% and $mAP_{0.5}$ by 48.4%. Taken together, it can be concluded that our improvement makes the model outperform the original YOLOv5s-SEG model in terms of segmentation performance and has a lighter weight.

Table 2: Ablation experiments with the lightweight semantic segmentation network YOEC-SEG

Names	Parameters	Size(MB)	Pixel Accuracy	$mAP_{0.5}$	$mAP_{0.5:0.95}$	GFLOPs	Inference Time(ms)
Baseline(Yolov5s-seg)	7.61M	15.6	0.673	0.532	0.319	26.4	42.3
Baseline-ghostconv-cbamc3(n)	6.89M	14.1	0.869	0.804	0.494	24.6	22.0
Baseline-ghostconv-cbamc3(b+n)	6.59M	13.6	0.831	0.753	0.448	25.5	18.5
Baseline-ghostconv-simsppf-cbamc3(n)	**6.23M**	**12.8**	0.863	0.79	0.478	**24.1**	**15.2**
Baseline-ghostconv-simsppf-cbamc3(b+n)	6.93M	14.2	0.869	0.799	0.49	24.7	16.6

3.4 Improved SLAM experimental results

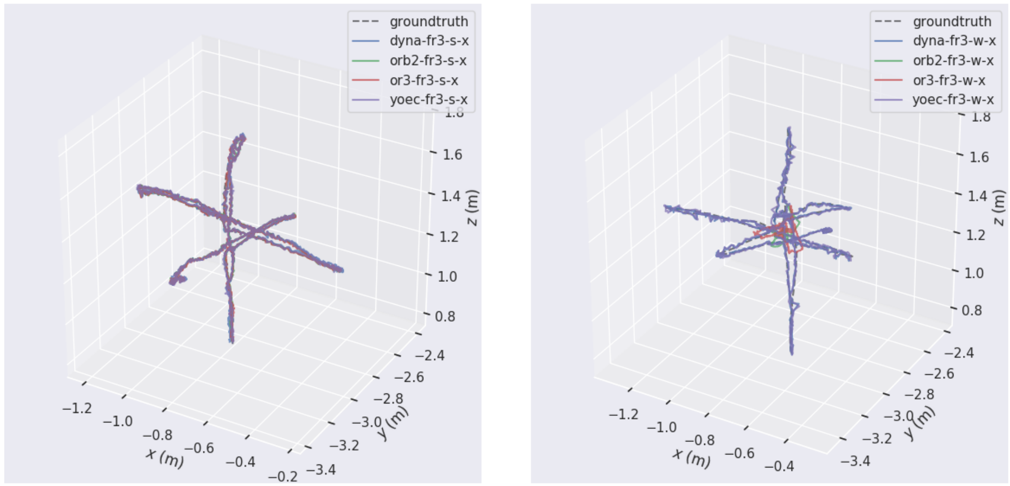
In this paper, we are more interested in the RMSE and S.D. (Standard Deviation) data as they give a better indication of the robustness and stability of the system.

Fig.5 shows the trajectory plots of the various algorithms in the high dynamic fr3_walking_xyz sequence and the low dynamic sequence fr3_sitting_xyz, respectively. We can see that the four SLAM systems perform in the low dynamic sequence fr3_sitting_xyz scenario. This experiment shows the trajectory maps generated by ORB-SLAM2, ORB-SLAM3, DYNA-SLAM, and YOEC-DSLAM algorithms in the high dynamic sequence fr3_walking_xyz scenario and the low dynamic sequence fr3_sitting_xyz scenario with the corresponding groundtruth in the 3D plane using the EVO tool. In the low dynamic sequence fr3_sitting_xyz scenario, the four SLAM systems perform more stable and accurate. In the high dynamic sequence fr3_walking_xyz scenario, the ORB-SLAM2 and ORB-SLAM3 trajectories exhibit large errors. DYNA-SLAM and YOEC-DSLAM are more stable relative to groundtruth. This indicates that our algorithm, YOEC-DSLAM, is able to estimate the trajectories more accurately in highly dynamic scenarios.

This experiment also uses EVO to analyse the data on the trajectories and generates an error box plot to evaluate the performance of each algorithm in depth. Fig.6 (a) visualises the performance of the four algorithms in the low dynamic scenario does not differ much and all of them obtain a good performance. Fig.6 (b) represents the results for high dynamic scenarios, where YOEC-DSLAM and DYNA-SLAM perform much better than the ORB-SLAM2 and ORB-SLAM3 algorithms, which fully demonstrates the importance of incorporating dynamic feature point screening for high dynamic scenarios.

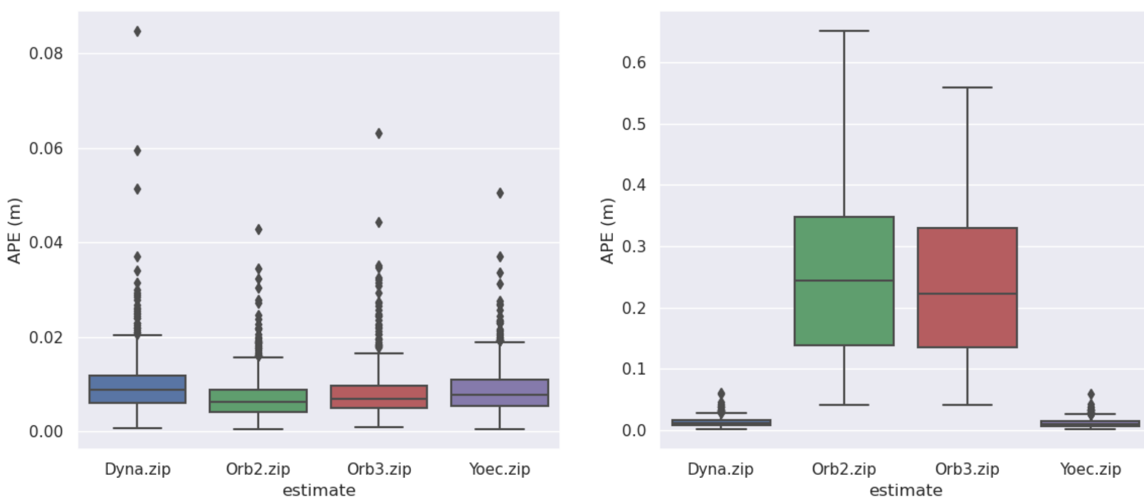
In real-world application scenarios, real-time performance is one of the key metrics for evaluating the performance of SLAM systems. To this end, we test the time consumed by the YOEC-DSLAM system in processing and tracking each image frame. We calculate the mean tracking time and median tracking time, which together reflect the real-time level of the system. In order to evaluate the performance of the YOEC-DSLAM system more comprehensively, we compare it with the existing DYNA-SLAM system. The test data are obtained from the indoor dynamic scene dataset TUM, and the experimental results are shown in Tab. 3.

The Table 3 shows that there is an improvement in processing time per frame in every sequence. In low dynamic sequences real time can be improved up to 94.7% and in high dynamic sequences real time can be improved up to 89.7%. The final result is an average mean tracking time improvement of 82.1% and an average median tracking time improvement of 82.7%.



(a) (b)

Figure. 5. Comparison of trajectory plots for different SLAM algorithms in dynamic different scenes: (a) Low dynamic scene: fr3_sitting_xyz sequence. (b) High dynamic scene: fr3_walking_xyz sequence.



(a) (b)

Figure. 6. Error box diagrams of DYNA-SLAM, ORBSLAM2, ORB-SLAM3, and YOEC-DSLAM in different scene: (a) fr3_sitting_xyz sequence. (b) fr3_walking_xyz sequence.

4. CONCLUSION

The YOEC-DSLAM system demonstrates its advantages when dealing with highly dynamic scenarios, achieving more efficient semantic segmentation and significantly improving real-time performance while maintaining a lightweight compared to ORB-SLAM2. This improvement is attributed to the optimisation of the YOLOv5s-SEG backbone network, including the introduction of the GhostConv and SimSPPF modules and the use of the CBAMC3 module in the Neck part, which effectively reduces the number of parameters and computational complexity of the model. In dynamic scenarios, the YOEC-DSLAM system combines semantic segmentation and multi-view geometry methods to effectively identify and filter dynamic feature points, improving the accuracy and robustness of map construction. Although the system performance is similar to that of ORB-SLAM2 in static or low-dynamic scenarios, its lightweight and real-time advantages are not fully utilised.

Table 3: Real-time comparison with DYNA-SLAM system and improved efficiency

Sequences		DYNASLAM		YOEC-DSLAM		Improvement	
		mean tracking time	median tracking time	mean tracking time	median tracking time	mean tracking time	median tracking time
TUM	fr3_walking_xyz	2.41236	2.42698	0.247879	0.253558	89.70%	89.60%
	fr3_walking_static	2.35420	2.35668	0.523183	0.498657	77.80%	78.80%
	fr3_walking_rpy	2.41976	2.42806	0.263568	0.267259	89.10%	90.00%
	fr3_sitting_xyz	2.40981	2.41736	0.438491	0.385913	81.80%	84.00%
	fr3_sitting_static	2.36182	2.37568	0.125999	0.113286	94.70%	95.20%

The YOEC-DSLAM system is a approach optimised for specific application scenarios and is particularly suitable for SLAM tasks in highly dynamic environments. However, for static or low-dynamic scenarios, the system did not demonstrate significant improvements. Future research could explore ways to better balance performance in different dynamic environments to improve the overall suitability of the system.

ACKNOWLEDGMENTS

The research is supported by the Cooperative Project between universities in Chongqing and affiliated institutes of the Chinese Academy of Sciences (No. HZ2021011), Action plan for quality development of Chongqing University of Technology graduate education (Grant No. gzlcx20232131) and Science and Technology Research Program of Chongqing Municipal Education Commission (Grant No. KJQN202201147). Postdoctoral Science Foundation Program of Chongqing Science and Technology Bureau (No. CSTB2022NSCQ-BHX0674).

REFERENCES

[1] Saijie Zhang and Huailin Zhao. Research and implementation of cooperative control for ros mobile robot. 2024.

[2] Miguel A Ferreira, Luís C Moreira, and António M Lopes. Autonomous navigation system for a differential drive mobile robot. *Journal of Testing and Evaluation*, 52(2):841–852, 2024.

[3] Wanni Arachchige Heshani Sandanika, Supun Hansaka Wishvajith, Sahan Randika, Deshitha Adeeshan Thennakoon, Samantha Kumara Rajapaksha, and Vishan Jayasinghearachchi. Ros-based multi-robot system for efficient indoor exploration using a combined path planning technique. *Journal of Robotics and Control (JRC)*, 5(5):1241–1260, 2024.

[4] Han Lei, Baoming Wang, Zuwei Shui, Peiyuan Yang, and Penghao Liang. Automated lane change behavior prediction and environmental perception based on slam technology. *arXiv preprint arXiv:2404.04492*, 2024.

[5] Iftekar Patel, Makarand Kulkarni, and Ninad Mehendale. Review of sensor-driven assistive device technologies for enhancing navigation for the visually impaired. *Multimedia Tools and Applications*, 83(17):52171–52195, 2024.

[6] Yu-Lin Zhao, Yi-Tian Hong, and Han-Pang Huang. Comprehensive performance evaluation between visual slam and lidar slam for mobile robots: Theories and experiments. *Applied Sciences*, 14(9):3945, 2024.

[7] Qiang Ji, Zikang Zhang, Yifu Chen, and Enhui Zheng. Drv-slam: An adaptive real-time semantic visual slam based on instance segmentation toward dynamic environments. *IEEE Access*, 12:43827–43837, 2024.

[8] Abhijit Kundu, K Madhava Krishna, and Jayanthi Sivaswamy. Moving object detection by multi-view geometric techniques from a single camera mounted robot. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4306–4312. IEEE, 2009.

[9] Danping Zou and Ping Tan. Coslam: Collaborative visual slam in dynamic environments. *IEEE transactions on pattern analysis and machine intelligence*, 35(2):354–366, 2012.

[10] Runzhi Wang, Wenhui Wan, Yongkang Wang, and Kaichang Di. A new rgb-d slam method with moving object detection for dynamic indoor scenes. *Remote Sensing*, 11(10):1143, 2019.

[11] Berta Bescos, José M Fácil, Javier Civera, and José Neira. Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4):4076–4083, 2018.

[12] Chao Yu, Zuxin Liu, Xin-Jun Liu, Fugui Xie, Yi Yang, Qi Wei, and Qiao Fei. Ds-slam: A semantic visual slam towards dynamic environments. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 1168–1174. IEEE, 2018.

[13] Fangwei Zhong, Sheng Wang, Ziqi Zhang, and Yizhou Wang. Detect-slam: Making object detection and slam mutually beneficial. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 1001–1010. IEEE, 2018.

[14] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 386–397, Feb 2020.

[15] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

[17] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017.

[18] Wei Tan, Haomin Liu, Zilong Dong, Guofeng Zhang, and Hujun Bao. Robust monocular slam in dynamic environments. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 209–218. IEEE, 2013.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[20] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012.